IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Fintan Ryan	§ §	Examiner: Boutah, Alina A.
	§	Group Art Unit: 2143
Serial No. 10/077,520	§ §	Atty. Dkt. No.: 5181-78701
Filed: February 15, 2002	§ §	
For: System and Method for Batch Tuning Intelligent Devices	§ § §	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

In response to the Notice of Panel Decision mailed November 14, 2007, Appellants present this Appeal Brief. **No extension of time is due since this Appeal Brief is filed within one month of the mailing date of the Notice of Panel Decision.** Appellant respectfully requests that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 012619/0910, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-72 are pending in the application and stand finally rejected. The rejection of claims 1-72 is being appealed. A copy of claims 1-72 is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method for generating a batch configuration document for an intelligent device. (*See, e.g.,* p. 5, lines 12-15; p. 17, lines 8-10.)

The method includes accessing a plurality of configuration files on the intelligent device. (*See, e.g.,* p. 17, line 30 – p. 18, line 3; FIG. 4, intelligent device 200 and configuration files 206; p. 32, lines 8-10; and FIG. 12, element 300.) Each of configuration files includes configuration information for one of a plurality of software components of the intelligent device. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.)

The method also includes generating the batch configuration document from the plurality of configuration files. (*See, e.g.,* p. 5, lines 11-15; p. 18, lines 1-3; FIG. 4, configuration files 206 and batch configuration document 212; p.32 lines 6-12; and FIG. 12, element 304.) The batch configuration document includes the configuration information for the plurality of software components of the intelligent device. (*See, e.g.,* p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

After the batch configuration document is generated, it is accessible for use in configuring the software components of the intelligent device whose configuration files were used in generating the batch configuration document. (*See, e.g.,* p. 33, lines 16-19; and FIG. 14, element 350.)

Independent claim 16 is directed to a method for configuring a plurality of software components of an intelligent device. (*See, e.g.*, Abstract; and p. 33 lines 16-18.)

The method includes accessing a batch configuration document that includes configuration information for the software components of the intelligent device. (*See, e.g.,* p. 33, lines 18-19, FIG. 14, element 350; p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

The method also includes applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device. (*See, e.g.,* p. 33 lines 21-28; and FIG. 14, element 354.)

Each of the configuration files includes configuration information for one of the plurality of software components of the intelligent device. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.)

Independent claim 33 is directed to a method for configuring intelligent devices. (*See, e.g.*, Abstract; and p. 33 lines 16-18.)

The method includes generating a batch configuration document from a plurality of configuration files on a first intelligent device. (*See, e.g.,* p. 27, lines 8-9 and 13-15.) Each of the configuration files includes configuration information for a software component of the intelligent device. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.) The batch configuration document includes the configuration information from the plurality of configuration files. (*See, e.g.,* p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

The method also includes configuring one or more software components of a second intelligent device using the batch configuration document generated on the first intelligent device. (*See, e.g.,* p. 9, lines 7-9; FIG. 9, batch configuration document 212; and p. 27, lines 9-11.)

Independent claim 41 is directed to a method for generating a batch configuration document for a plurality of intelligent devices. (*See, e.g.,* p. 5, lines 12-15; p. 9, lines 13-14; p. 17, lines 8-10; and p. 30, lines 2-4.)

The method includes accessing one or more configuration files on each of the intelligent devices. (*See, e.g.,* p. 17, line 30 – p. 18, line 3; FIG. 4, intelligent device 200 and configuration files 206; p. 30, lines 5-9; p. 32, lines 8-10; and FIG. 12, element 300.) Each of the configuration files includes configuration information for one of one or more software components of the intelligent device. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5; p. 9, lines 14-19; and p. 28, lines 5-9.)

The method includes generating the batch configuration document from the configuration files on each of the intelligent devices. (*See, e.g.,* p. 9, lines 14-19; and p. 28, lines 5-9.) The batch configuration document includes the configuration information for the software components of each of the intelligent devices. (*See, e.g.,* p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

After the batch configuration document is generated, it is accessible for use in configuring the intelligent devices whose configuration files were used in generating the batch configuration document. (*See, e.g.,* p. 9, lines 19-21.)

Independent claim 48 is directed to an intelligent device that includes a processor, a plurality of software components, a plurality of configuration files, and a memory. (*See, e.g.,* p. 14, lines 15-17; FIG. 4, software components 202, configuration files 206.)

Each of the configuration files is associated with one of the software components, and includes configuration information for its associated component. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.)

The memory is operable to store program instructions executable by the processor to open each of the plurality of configuration files to access the configuration information for the component associated with the configuration file. (*See, e.g.,* FIG. 4, generator script 214, intelligent device 200, and configuration file 206; p. 17, line 30 – p. 18, line 3; p. 32, lines 8-10; p. 35, lines 11-21; and FIG. 12, element 300.)

The program instructions are also executable by the processor to generate a batch configuration document from the configuration information accessed from each of the configuration files. (*See, e.g.,* p. 5, lines 12-15; p. 17, lines 8-10; and p. 18, lines 1-3.) The batch configuration document includes the configuration information from each of the configuration files. (*See, e.g.,* p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

After the batch configuration document is generated, it is accessible for use in configuring the software components in the intelligent device. (*See, e.g.,* p. 33, lines 16-19; and FIG. 14, element 350.)

Independent claim 56 is directed to an intelligent device that includes a processor, a plurality of software components, a plurality of configuration files, and a memory. (*See, e.g.,* p. 14, lines 15-17; and FIG. 4, software components 202, configuration files 206.)

Each of the configuration files is associated with one of the plurality of software components and includes configuration information for its associated component. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.)

The memory is operable to store program instructions executable by the processor to open a batch configuration document that includes configuration information for the software components of the intelligent device. (*See, e.g.,* p. 33, lines 18-19, FIG. 14, element 350; p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

The program instructions are also executable by the processor to apply the configuration information from the batch configuration document to the configuration files on the intelligent device. (*See, e.g.,* p. 33 lines 21-28; and FIG. 14, element 354.)

Independent claim 66 is directed to a tangible, computer-accessible storage medium comprising program instructions. (*See, e.g.,* FIG. 4, generator script 214; p. 17, lines 26-30; and p. 35, lines 11-21.)

The program instructions are computer-executable to implement accessing a plurality of configuration files on an intelligent device. (*See, e.g.,* p. 17, line 30 – p. 18, line 3; FIG. 4, intelligent device 200 and configuration files 206; p. 32, lines 8-10; and FIG. 12, element 300.) Each of the configuration files includes configuration information for one of a plurality of software components of the intelligent device. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.)

The program instructions are also computer-executable to implement generating a batch configuration document from the plurality of configuration files that includes the configuration information for the software components of the intelligent device. (*See, e.g.,* p. 5, lines 12-15; p. 17, lines 8-10 and lines 26-30; FIG. 4, generator script 214, configuration files 206, and batch configuration document 212; p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.

After the batch configuration document is generated, it is accessible for use in configuring the software components of the intelligent device whose configuration files were used in generating the batch configuration document. (*See, e.g.,* p. 33, lines 16-19; and FIG. 14, element 350.)

Independent claim 70 is directed to a tangible, computer-accessible storage medium comprising program instructions. *(See, e.g.,* FIG. 4, tuner script 216; p. 19, lines 13-18; and p. 35, lines 11-21.)

The program instructions are computer-executable to implement accessing a batch configuration document. (See, e.g., FIG. 11, intelligent devices 200 and batch configuration documents 212; p. 19, lines 18-19; and p. 31, lines 10-19.) The batch configuration document includes configuration information for a plurality of software components of an intelligent device. (*See, e.g.,* p. 19, lines 1-11; p. 28, lines 3-6; and FIG. 5.)

The program instructions are also computer-executable to implement applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device. (*See, e.g.,* p. 19, lines 2-21; and p. 31, lines 19-23.) Each of the configuration files includes configuration information for one of the software components of the intelligent device. (*See, e.g.,* p. 2, lines 22-26; and p. 5, lines 4-5.)

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-5, 7-18, 23-56, 60-66, 68, 69 and 72 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Horman (U.S. Patent 6,785,706) in view of Mossman (U.S. Publication 2002/0124061).

2. Claims 6, 19-22, 57-59, 67 and 71 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Horman in view of Mossman and in further view of Shafron et al. (U.S. Publication 2003/0014479).

VII. ARGUMENT

First ground of rejection:

Claims 1-5, 7-18, 23-56, 60-66, 68, 69 and 72 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Horman (U.S. Patent 6,785,706) in view of Mossman (U.S. Publication 2002/0124061). Appellant traverses this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 9, 10, 41, 48, 52, 66, and 68:

1. The cited art clearly fails to teach or suggest ***accessing a plurality of configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of a plurality of software components of the intelligent device.***

The Examiner's citations in Horman (abstract; column 1, line 66 – column 2, line 6) describe an administrative control server configured to change the configurations of administered servers according to synchronization instructions generated from configuration information stored on the administrative control server. **There is nothing in these citations, or elsewhere in the combination of Horman and Mossman, to teach or suggest accessing a plurality of configuration files on the intelligent device for which a batch configuration document is generated, as recited in claim 1. Also, there is nothing in the combination of Horman and Mossman to teach or suggest one or more configuration files, each including configuration information for one of a plurality of software components of the intelligent device, as recited in claim 1.** The configuration information stored on the administrative control server of Horman instead includes items describing server configurations, such as which administered servers are in the environment, the group each administered server belongs to, and which version of end-user applications an administered server is running, not information for configuring individual software components on an intelligent device.

The Examiner admits, in the Office Action dated March 20, 2006, that Horman does not explicitly teach the configuration files being accessed on the intelligent device itself. The Examiner submits that Mossman teaches this limitation. However, in Mossman, data is collected from the user (not accessed from a configuration file on an intelligent device) and stored on the server. For example, configuration documents 132 are located on the server, and configuration parameters relations database 64 and parameter values database 60 are located on configuration system 10, which is on the server side of system 100. Therefore, the Examiner's additional citations in Mossman do not overcome the deficiency of Horman in teaching or suggesting this limitation. In the Response to Arguments section of the Office Action dated August 23, 2007, the Examiner cites Horman as defining what is in the configuration file. However, this citation describes what is included in a database definition and does not teach a configuration file. Appellant again asserts that data obtained from a user through a graphical user interface (Mossman) and stored on a server is clearly not analogous to accessing configuration files on an intelligent device, as recited in claim 1.

2. The cited art clearly fails to teach or suggest *generating the batch configuration document from the plurality of configuration files, wherein the batch configuration document includes the configuration information for the plurality of software components of the intelligent device.*

The Examiner cites Horman (column 5, lines 43-55) as teaching this limitation. However, this citation describes generating synchronization instructions based on which batches (i.e., sets) of synchronization scripts apply to each administered server. There is nothing in this citation, or elsewhere in the combination of Horman and Mossman, that teaches or suggests that these synchronization scripts or synchronization instructions include configuration information for the plurality of software components of the intelligent device, as recited in claim 1. In the Response to Arguments section of the Office Action dated August 23, 2007, the Examiner states, "The batch file includes end-user application as well as database definition... The end-user application comprises software component, as well known in the art." However, a batch file that includes an end-user application or a software component has no bearing on what is actually recited in claim 1, ***generating the batch configuration document from the plurality of configuration files that include configuration information software components, wherein the batch configuration document includes the configuration information for the plurality of software components of the intelligent device.*** There is nothing in the cited references that teaches or suggests that any of the "batches" of Horman are generated from configuration files on a device targeted for configuration using the batch, as in Appellant's claim. In addition, there is nothing in Horman that describes that one of the "batches" of Horman includes configuration information for software components on that device that was included in the configuration files from which they were generated (i.e., those on a device targeted for configuration using the batch), as required by claim 1.

3. The cited art clearly fails to teach or suggest *the batch configuration document is accessible for use in configuring the plurality of software components of the intelligent device whose configuration files were used in said generating the batch configuration document.*

The Examiner cites Horman (column 5, lines 43-55 and column 5, line 66 – column 6, line 5) as teaching this limitation. However, as discussed above, neither of these citations teaches or suggests a batch configuration document for configuring a plurality of software components of an intelligent device whose configuration files were used in generating the batch configuration document. There is nothing in the combination of Horman and Mossman that teaches or suggests using such a batch configuration document to configure a plurality of software components on the same intelligent device whose configuration files were used in generating a batch configuration document. The Examiner's citation does not teach "generating batches" at all, much less in the manner recited in Appellant's claims. Horman merely teaches the use of scripts known as "batches" for database definitions and data, without describing their generation.

4. The Examiner has not established a proper reason to combine the references.

The Examiner submits that at the time the invention was made, one of ordinary skill in the art would have been motivated to access configuration files on an intelligent device in order to configure a plurality of parameters of a target device, therefore optimizing the device for its intended use. **The Examiner's assertion is completely unsupported by any evidence of record.** Mossman does not teach or suggest this limitation. Furthermore, Horman purposefully changes the configurations of its administered servers without this feature. In fact, the Examiner admits that "Horman does not explicitly teach the configuration files being accessed on the intelligent device itself." As shown above, the feature is not clearly not taught by Mossman either, whether considered alone or in combination with Horman. Thus, the Examiner's reason to combine the references is based on teachings that are not actually present in the references. Therefore, the Examiner has failed to provide sufficient motivation to combine the references. Moreover, even if the references were combined, the resultant combination would not produce Appellant's claimed invention, as shown above.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested.

Appellant's arguments above regarding claim 1 apply also to claims 41, 48, and 66, which recite similar limitations and were rejected under the same rationale as claim 1.

Claims 16, 23, and 24:

1. The Examiner failed to address the differences between claims 1 and 16; therefore, the Examiner has failed to state a *prima facie* rejection of claim 16.

The Examiner rejected claim 16 under the same rationale as claim 1. **Appellant asserts that both the scope and specific limitations of claim 1 and claim 16 differ.**

For example, claim 1 recites, "A method for generating a batch configuration document for an intelligent device." It includes, "accessing a plurality of configuration files on the intelligent device, wherein each of the one or more configuration files include configuration information for one of a plurality of software components of the intelligent device." By comparison, claim 16 recites, "A method for configuring a plurality of software components of an intelligent device." It includes, "accessing a batch configuration document, wherein the batch configuration document comprises configuration information for the plurality of software components of the intelligent device." **Appellant asserts that both the scope and specific limitations of the methods of claims 1 and 16 are clearly different, and that the Examiner did not address these differences.**

2. The cited art clearly fails to teach or suggest *accessing a batch configuration document, wherein the batch configuration document comprises configuration information for the plurality of software components of the intelligent device; and applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of the plurality of software components of the intelligent device* (as in claim 16).

In remarks regarding claim 1, the Examiner cites Horman (column 5, lines 43-55) as teaching Appellant's batch configuration file. However, this citation describes generating synchronization instructions based on which batches (i.e., sets) of synchronization scripts apply to each administered server. There is nothing in this citation, or elsewhere in the combination of Horman and Mossman, that teaches or suggests that each of these synchronization scripts or synchronization instructions include configuration information for a plurality of software components of an intelligent device, as recited in claim 16. **Instead, in Horman, a group batch is described as being associated with a single version of a single end-user application.**

For at least the reasons above, the rejection of claim 16 is not supported by the cited art and removal thereof is respectfully requested.

Claim 70:

1. The Examiner has failed to state a *prima facie* rejection of claim 70.

Appellant notes that claim 70 is not included in the listing of claims rejected under the first or second ground of rejection. The Examiner includes in his remarks, "Claims 70 and 72 are similar to claims 41 and 47, respectively, therefore are rejected under the same rationale." Therefore, Appellant assumes that the omission of claim 70 in the listing of claims rejected under the first ground of rejection was an inadvertent oversight by the Examiner, and that claim 70 was meant to be included in this listing. **Appellant also notes that the scope of claim 70 (which recites limitations similar to those of claim 16) differs from that of claim 41 (which is similar to claim 1 and was rejected under the same rationale as claim 1).** For example, claim 70 recites accessing a batch configuration document, wherein the batch configuration document comprises configuration information. Claim 41 includes no such limitation. Claim 70 recites, applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device. Claim 41 includes no such limitation.

Because the Examiner has failed to address the differences between claim 70 and claim 41, the Examiner has failed to state a *prima facie* rejection of claim 70.

2. The cited art clearly fails to teach or suggest *accessing a batch configuration document, wherein the batch configuration document comprises configuration information for a plurality of software components of an intelligent device; and applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of the plurality of software components of the intelligent device.*

In remarks regarding claim 1, the Examiner cites Horman (column 5, lines 43-55) as teaching Appellant's batch configuration file. However, this citation describes generating synchronization instructions based on which batches (i.e., sets) of synchronization scripts apply to each administered server. There is nothing in this citation, or elsewhere in the combination of Horman and Mossman, that teaches or suggests that each of these synchronization scripts or synchronization instructions include configuration information for a plurality of software components of an intelligent device, as recited in claim 70. **Instead, in Horman, a group batch is described as being associated with a single version of a single end-user application.**

For at least the reasons above, the rejection of claim 70 is not supported by the cited art and removal thereof is respectfully requested.

Claim 33 and 37:

1. **The cited art clearly fails to teach or suggest *generating a batch configuration document from a plurality of configuration files on a first intelligent device* and *configuring one or more software components of a second intelligent device using the batch configuration document generated on the first intelligent device.***

As discussed above, regarding claim 1, Horman in view of Mossman fails to teach or suggest generation of a batch document for configuring software components of an intelligent device. There is also nothing in Horman or Mossman, or the combination thereof, that teaches or suggests generating the batch configuration document on a first intelligent device and using it to configure software components on a second intelligent device, as recited in claim 33. In the Response to Arguments section of the Office Action mailed August 23, 2007, the Examiner asserts, "[t]he control server creates a synchronization script. The synchronization script is used to configure the administered servers." **A synchronization script does not teach or suggest generating a batch configuration document from a plurality of configuration files on a first intelligent device and configuring one or more software components of a second intelligent**

<u>device</u> **using the batch configuration document generated on the first intelligent device**. Appellant asserts that Horman and Mossman clearly do not teach this limitation.

For at least the reasons above, the rejection of claim 33 is not supported by the cited art and removal thereof is respectfully requested.

<u>**Claims 56 and 60:**</u>

1. The cited art clearly fails to teach or suggest *a plurality of software components* and *a plurality of configuration files, wherein each of the plurality of configuration files is associated with one of the plurality of software components, and wherein each of the plurality of configuration files includes configuration information for its associated component.*

The Examiner cites Horman (abstract; column 1, line 66 – column 2, line 6) as teaching these limitations. However, as discussed above, Horman does not describe configurable software components of an intelligent device, with associated configuration files including configuration information for the software components.

2. The cited art clearly fails to teach or suggest *the batch configuration document comprises configuration information for the plurality of software components of the intelligent device* and *apply the configuration information from the batch configuration document to the plurality of configuration files on the intelligent device.*

The Examiner refers to Horman, figure 5A, as teaching these limitations. However, figure 5A describes a synchronization procedure in which <u>parameters of sync scripts</u> may be replaced with new values before being downloaded and executed on an administered server to synchronize it with its group. There is nothing in figure 5A, or elsewhere in Horman, that teaches or suggests <u>applying configuration information</u> from a batch document <u>to configuration files</u> located <u>on the intelligent device</u>, as recited in claim

56. In the Office Action dated March 20, 2006, the Examiner admitted that Horman does not explicitly teach the configuration files being accessed on the intelligent device itself. The Examiner submits that Mossman teaches this limitation in the abstract and FIG. 5. However, the Examiner is incorrect. As discussed above, FIG. 5 and its accompanying description clearly depict configuration documents 132 on the server, not on an intelligent device to be configured. Similarly, FIG. 3 of Mossman illustrates configuration parameters relations database 64 and parameter values database 60 on configuration system 10, which is on the server side of system 100 (see FIG. 5). As illustrated in FIGs. 3 and 5, data is collected from the user (not accessed from a configuration file on an intelligent device) and stored in the parameters values database 60 on the server. Information displayed for the user comes from configuration parameter relations database 64, which is also on the server. Therefore, the Examiner's additional citations in Mossman do not overcome the deficiency of Horman in teaching or suggesting applying the configuration information from the batch configuration document *to the plurality of configuration files on the intelligent device*, as recited in claim 56.

3. The Examiner has not established a proper reason to combine the references.

The Examiner submits that at the time the invention was made, one of ordinary skill in the art would have been motivated to access configuration files on an intelligent device in order to configure a plurality of parameters of a target device, therefore optimizing the device for its intended use (Mossman [0005]). Appellant asserts, however, that Mossman does not teach or suggest this limitation. Furthermore, Horman changes the configurations of its administered servers without this feature. Therefore, the Examiner has failed to provide sufficient motivation to combine the references.

For at least the reasons above, the rejection of claim 56 is not supported by the cited art and removal thereof is respectfully requested.

Claim 2:

1. **The cited art clearly fails to teach or suggest** ***wherein said accessing the plurality of configuration files and said generating the batch configuration document are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for selecting the plurality of configuration files to be accessed and one or more executable instructions for performing said generating the batch configuration document.***

The Examiner cites Horman (abstract; col. 1, line 66 – column 2, line 6; FIG. 1) as teaching these limitations. This passage describes "the control server can change the configurations of the administered servers to a new desired configuration for each administered server where the new configuration is in synchronization with the configuration of each of said other administered server, by generating synchronization instructions from parameterized synchronization scripts stored in the control database." It does not describe generating the batch configuration document of Appellant's claims by executing a script on the intelligent device. The synchronization scripts do not meet the limitations recited in Appellant's claims for the batch configuration document of the present invention, nor are they generated on the intelligent device from which configuration information for multiple software components were accessed, but on the control server.

For at least the reasons above, the rejection of claim 2 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 3:

1. **The cited art clearly fails to teach or suggest** ***prior to said accessing the plurality of configuration files, configuring the plurality of software components of the intelligent device, wherein said configuring the plurality of software components sets the configuration information in the plurality of configuration files.***

The Examiner again cites Horman (abstract; col. 1, line 66 – column 2, line 6) as teaching these limitations. However, as discussed above, this passage describes a control server generating synchronization instructions for synchronizing administrative servers using synchronization scripts stored in a control database. It does not teach or suggest anything about <u>configuring software components</u> on one of the administrative servers <u>prior to accessing configuration files on those servers</u> (from which a batch configuration document is to be generated), as in claim 3.

For at least the reasons above, the rejection of claim 3 is unsupported by the cited art and removal thereof is respectfully requested.

<u>Claim 4:</u>

1. The cited art clearly fails to teach or suggest *transferring the batch configuration document to another intelligent device for use in configuring one or more software components of the other intelligent device.*

The Examiner cites Horman (col. 2, lines 28-41) as teaching this limitation. This passage describes executing a script of commands at each database server to optimize memory allocation configuration. It does not describe the batch configuration document of Appellant's claims (i.e., one generated on an intelligent device from configuration files on that device corresponding to software components on that device), much less transferring such a document to another device for use in configuring software components on the other device.

For at least the reasons above, the rejection of claim 4 is unsupported by the cited art and removal thereof is respectfully requested.

<u>Claim 5:</u>

1. **The cited art clearly fails to teach or suggest *wherein the batch configuration document further includes configuration information for one or more software components of one or more other intelligent devices.***

The Examiner again cites Horman (abstract; col. 1, line 66 – col. 2, line 6; and FIG. 1) as teaching this limitation. However, Horman does not describe the batch configuration document of Appellant's claims, nor that such a batch configuration document generated on one device from the configuration files associated with software components on that device includes configuration information for software components on another device, as recited in claim 5. Instead, Horman describes synchronization script templates on a control server used to configure administrative servers.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 7:

1. **The cited art clearly fails to teach or suggest *configuring one or more of the plurality of software components of the intelligent device using the batch configuration document, wherein said configuring comprises applying the configuration information from the batch configuration document to one or more of the plurality of configuration files, wherein each of the one or more of the plurality of configuration files is associated with one of the one or more of the plurality of software components of the intelligent device.***

The Examiner cites Horman (col. 5, lines 43-55; and col. 5, line 64 – col. 6, line 5) as teaching these limitations. These passages describe the database definitions may be set up using scripts known as batches and that group batches are associated with particular application versions. It does not describe configuring databases using the batch configuration document of Appellant's claims, nor applying configuration information from such a batch configuration document (generated on one intelligent device from configuration files on that device associated with software components on that device) to the configuration files on that device.

For at least the reasons above, the rejection of claim 7 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 8:

1. **The cited art clearly fails to teach or suggest** ***configuring the one or more of the plurality of software components of the intelligent device further comprises initializing each of the one or more of the plurality of software components, wherein said initializing uses the configuration information from the one or more configuration files associated with the particular component.***

The Examiner cites Horman, column 7, lines 32-36 as teaching this limitation. However, this passage describes **modeling** an initial deployment by creating an install image based on the model office administered server. This has nothing to do with configuring a plurality of software components of an intelligent device, nor with initializing the software components using configuration information from associated configuration files. In the Response to Arguments section of the Office Action mailed August 23, 2007, the Examiner states, "It is well known in the art that in order to execute a software, it must be initialized." The Examiner has clearly not addressed all of the limitations in the Appellant's claim, such as **configuring a plurality of software components of an intelligent device further comprises initializing each of the one or more of the plurality of software components, wherein said initializing uses the configuration information from the one or more configuration files associated with the particular component.**

For at least the reasons above, the rejection of claim 8 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 11:

1. **The cited art clearly fails to teach or suggest** ***wherein the plurality of software components includes software drivers for hardware components.***

The Examiner cites Horman (col. 6, line 57 – col. 7, line 2) as teaching this limitation. **While this passage includes the word "hardware," it has absolutely nothing to do with the limitations of claim 11** regarding software drivers for hardware components. Instead, this passage includes the following, "The hardware on which the DBMS and the business application run can be any laptop or desktop computer on which any of the supported operating systems is running."

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 12, 38, 45, 53, and 61:

1. **The cited art clearly fails to teach or suggest** ***wherein at least one of the plurality of configuration files includes operating system configuration information for the intelligent device.***

The Examiner again cites Horman (col. 6, line 57 – col. 7, line 2) as teaching this limitation. **While this passage includes the phrase "operating systems," it has absolutely nothing to do with the limitations of claim 12** regarding configuration files including operating system configuration information for an intelligent device. Instead, it includes the following, "The hardware on which the DBMS and the business application run can be any laptop or desktop computer on which any of the supported operating systems is running."

For at least the reasons above, the rejection of claim 12 is unsupported by the cited art and removal thereof is respectfully requested.

Appellant's arguments above regarding claim 12 apply also to claims 38, 45, 53, and 61, which recite similar limitations and were rejected under the same rationale as claim 12.

Claims 13-14, 27-28, 35, 46, 54, 62, 69, and 72:

1. The cited art clearly fails to teach or suggest *the batch configuration document is a markup language document* (as in claim 13).

The Examiner cites Mossman (paragraph 0091) as teaching this limitation. While this paragraph describes configuration documents that are static files stored in an extensible Markup Language (XML) format, these documents are not batch configuration documents, as recited in Appellant's claims. Instead, each is a configuration document containing instructions to manage a (single) configuration instance 80. There is nothing in this paragraph, or elsewhere in Mossman, that teaches or suggests a batch configuration document is a markup language document, such as XML, as recited in claim 13. The Examiner has not cited anything in Horman or Mossman that teaches or suggests this limitations.

2. The Examiner has not established a proper motivation to combine what is known in the art with the teachings of Horman and Mossman in his remarks regarding this or any other claim of the present invention, and there is nothing in these references or any other art of record to suggest such a combination.

In the Response to Arguments section of the Office Action mailed August 23, 2007, the Examiner states, "Mossman teaches configuration files being XML format. The use of batch files have been known in the art. It being a mark up language is a matter of design choice. Therefore, although not explicitly taught, this feature is obvious in the art." **Appellant asserts, however, that the Examiner has failed to provide a**

reason that it would be obvious to combine what is known in the art with the teachings of Horman and Mossman.

For at least the reasons above, the rejection of claims 13 is unsupported by the cited art and removal thereof is respectfully requested.

Similar remarks as those discussed above regarding claim 13 apply also to dependent claims 14, 27-29, 35-36, 46-47, 54-55, 62-63, 69, and 72, which include limitations similar to those of claim 13.

<u>**Claims 15, 29, 36, 47, 55, 63, 69, and 72:**</u>

1. The cited art clearly fails to teach or suggest *wherein the batch configuration document and the plurality of configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD)* (as in claim 15).

The Examiner again cites Mossman (paragraph 0091) as teaching this limitation. While this paragraph describes configuration documents that are static files stored in an extensible Markup Language (XML) format, these documents are not <u>batch configuration documents</u>, as recited in Appellant's claims. Instead, each is a configuration document containing instructions to manage a (single) configuration instance 80. There is nothing in this paragraph, or elsewhere in Mossman, that teaches or suggests a <u>batch configuration document</u> is a markup language document, such as XML, or the specific limitations of claim 15 regarding the use of XML for a batch configuration document, or conformity to an XML Document Type Definition (DTD). There is nothing in the cited references teaching batch configuration documents conforming to an XML DTD.

For at least the reasons above, the rejection of claims 15 is unsupported by the cited art and removal thereof is respectfully requested.

Similar remarks as those discussed above regarding claim 15 apply also to dependent claims 29, 36, 47, 55, 63, 69, and 72, which include limitations similar to those of claim 15.

Claim 17:

1. **The cited art clearly fails to teach or suggest** ***wherein said applying the configuration information from the batch configuration document to each of the one or more configuration files comprises replacing one or more current parameter values in the particular configuration file with new parameter values from the batch configuration document.***

The Examiner cites Horman (col. 2, lines 50-65) as teaching these limitations. First, Appellant again asserts that the cited art does not teach the batch configuration document of Appellant's claims. In addition, this passage describes replacing parameters in a parameterized script template when instantiating an actual script. It has nothing to do with applying configuration information in such a batch configuration document to configuration files associated with software components on an intelligent device by replacing parameter values in those configuration files with parameter value from such a batch configuration document.

For at least the reasons above, the rejection of claim 17 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 18:

1. **The cited art clearly fails to teach or suggest** ***wherein said accessing and said applying are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for accessing the batch***

configuration document and one or more executable instructions for selecting the one or more configuration files to be configured.

The Examiner again cites Horman (col. 2, lines 50-65) as teaching these limitations. However, as discussed above, this passage describes the instantiation of a script to be executed on a database server. It does not describe that this script includes the instructions for accessing and selecting recited in claim 18, much less that it accesses a batch configuration document, such as that of Appellant's claims, or applies the configuration information therein to selected configuration files on the intelligent device.

For at least the reasons above, the rejection of claim 18 is unsupported by the cited art and removal thereof is respectfully requested.

<u>**Claims 30, 39, and 64:**</u>

1. **The cited art clearly fails to teach or suggest *the method as recited in claim 16, further comprising rebooting the intelligent device after said applying the configuration information from the batch configuration document to the one or more configuration files, wherein said rebooting applies the configuration information from the one or more configuration files to one or more of the plurality of software components of the intelligent device.***

The Examiner relies on Mossman (paragraph 0153) to teach this limitation. However, this paragraph <u>teaches away</u> from this limitation, by describing that, "If these settings were established directly to the system 12 a system reboot would be required," but that "in the configuration system 10 of the present invention, the settings and the required reboot invocation are applied to <u>the virtual system 16</u>", rather than directly to the actual system 12. Furthermore, there is nothing in this paragraph or elsewhere that describes that <u>**rebooting** applies configuration information</u> from one or more configuration files <u>to one or more software components</u> of the intelligent device, as recited in claim 30. In the Response to Arguments section of the Office Action mailed August 23, 2007, the Examiner states, "It is well known in the computer art that in order for any change or configuration to take place, the computer must be rebooted." Appellants assert, however, that the Examiner has not cited anything in Horman or Mossman that teaches or suggests all of the specific limitations of claim 30.

2. The Examiner has not established motivation to combine the teachings of Horman and Mossman in his remarks regarding this claim of the present invention, and there is nothing in these references or any other art of record to suggest such a combination.

Appellant's arguments above regarding claim 30 apply also to claims 39 and 64, which recite limitations similar to those of claim 30 regarding rebooting.

Claims 31 and 65:

1. The cited art clearly fails to teach or suggest *initializing one or more of the plurality of software components of the intelligent device after said applying the configuration information from the batch configuration document to the one or more configuration files, wherein, in said initializing, each of the one or more of the plurality of software components is initialized using the configuration information from each of the one or more configuration files associated with the particular component*.

The Examiner relies on Mossman (paragraph 108) as teaching this limitation. This paragraph describes only that a web server has the responsibility to "verify that the system output interface 66 can be initialized on start-up". This has nothing to do with initializing one or more software components of an intelligent device after applying configuration information from a batch document to configuration files or with initializing each software component using configuration information from its associated configuration file. Nowhere does Horman, Mossman, or any combination thereof, teach or suggest this limitation.

2. The Examiner has not established a proper motivation to combine the teachings of Horman and Mossman in his remarks regarding this or any other claim of the present invention, and there is nothing in these references or any other art of record to suggest such a combination.

For at least the reasons above, the rejection of claim 31 is unsupported by the cited art and removal thereof is respectfully requested.

Appellant's arguments above regarding claim 31 apply also to claim 65, which recites limitations similar to those of claim 31 regarding initializing software components.

<u>**Claim 32:**</u>

1. The cited art clearly fails to teach or suggest *generating the batch configuration document on a different intelligent device prior to said accessing.*

The Examiner cites Horman (col. 8, lines 45-55) as teaching this limitation. This passage describes that an application version is set at the administered server and reported to the administrative control server prior to synchronization. **It has absolutely nothing to do with the limitations of claim 32.**

For at least the reasons above, the rejection of claim 31 is unsupported by the cited art and removal thereof is respectfully requested.

<u>**Claim 34:**</u>

1. The cited art clearly fails to teach or suggest *wherein said configuring comprises applying configuration information from the batch configuration document generated on the first intelligent device to one or more configuration files on the second device, wherein each of the one or more configuration files on the second*

intelligent device is associated with one of the one or more software components of the second intelligent device.

The Examiner again cites Horman (abstract; and col. 1, line 66 – col. 2, line 6) as teaching these limitations. This passage describes that a control server can change the configuration of administered servers by generating synchronization instructions from synchronization scripts. As discussed above, Horman does not teach the batch configuration document of Appellant's claims, or the application of such batch configuration documents in the manner recited in claim 34.

For at least the reasons above, the rejection of claim 34 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 40:

1. The cited art clearly fails to teach or suggest *storing the generated batch configuration document on a server, wherein the server is coupled to the second intelligent device via a network.*

The Examiner cites Horman (col. 1, line 60 – col. 2, line 6) as teaching this limitation. First Appellant's again assert that the cited art does not teach the batch configuration document of Appellant's claims (i.e., one generated as recited in claim 33). These passages of Horman do not teach storing such a batch configuration document on a server. Instead, they teach storing parameterized script templates and "batches" in an administrative control database, neither of which meets the limitations recited for Appellant's batch configuration document.

For at least the reasons above, the rejection of claim 40 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 42:

1. **The cited art clearly fails to teach or suggest *wherein the batch configuration document is further accessible for use in configuring other pluralities of intelligent devices.***

The Examiner cites Horman (abstract) as teaching this limitation. This passage describes parameterized synchronization script templates stored in the control database. However, as discussed above, these synchronization script templates do not meet the recited limitations for Appellant's batch configuration documents (e.g., regarding their generation or contents), much less that such a batch configuration document is accessible for use in configuring devices other than those that include the configuration files from which the batch configuration document was generated.

For at least the reasons above, the rejection of claim 42 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 43-44, and 51-52:

1. **The Examiner failed to address the differences between claims 43-44 and claims 3-4; therefore, the Examiner has failed to state a *prima facie* rejection of claims 43-44.**

The Examiner rejected claims 43-44 under the same rationale as claims 3-4. However, claims 43 and 44 are directed toward completely different subject matter than claims 3-4, none of which was addressed by the Examiner in remarks regarding claims 43-44. Therefore the rejection is improper. Appellant asserts that the Examiner has not cited anything in the references to teach or suggest these limitations.

Claims 51-52 recite limitations similar to those of claims 43-44 and were rejected under the same rationale as claims 43-44. However, as discussed above, the rejection of claims 43-44 is improper, as it was based on the rejection of dissimilar claims 3-4. Therefore the rejection of claims 51-52 is also improper.

For at least the reasons above, the rejection of claims 43-44 and 51-52 is unsupported by the cited art and removal thereof is respectfully requested.

<u>**Claim 49:**</u>

1. **The Examiner failed to address the differences between claim 49 and claim 42; therefore, the Examiner has failed to state a *prima facie* rejection of claim 49.**

The Examiner rejected claim 49 under the same rationale as claim 42. However, the scope of claim 49 differs from that of claim 42, and these differences were not addressed by the Examiner. For example, claim 49 recites *wherein the batch configuration document is further <u>transferable</u> to <u>another intelligent device</u>.* No such limitation is present in claim 42. In addition, claim 49 recites that the other intelligent device comprises *one or more software components similar to software components comprised in the plurality of software components of the intelligent device for use in configuring the other intelligent device.* No such limitation is present in claim 42. Instead, claim 42 recites, in its entirety, "The method as recited in claim 41, wherein the batch configuration document is further <u>accessible</u> for use in configuring other pluralities of intelligent devices." Since the Examiner failed to address these limitations of claim 49, the rejection is improper. **Appellant asserts that the Examiner has not cited anything in the references to teach or suggest these limitations, and that the references do not, in fact, teach these limitations.**

For at least the reasons above, the rejection of claim 49 is unsupported by the cited art and removal thereof is respectfully requested.

<u>**Second ground of rejection:**</u>

Claims 6, 19-22, 57-59, 67 and 71 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Horman in view of Mossman and in further view of Shafron et al. (U.S. Publication 2003/0014479) (hereinafter "Shafron"). Appellant traverses this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 6, 19-22, 50, 57-59, 67, and 71:

1. The cited art clearly fails to teach or suggest ***generating the batch configuration document comprises generating a Document Object Model (DOM) tree from each of the accessed configuration files, wherein the configuration information incorporated in the configuration document is accessed from the DOM trees generated from the plurality of configuration files.***

The Examiner relies on Shafron as teaching this limitation and cites paragraphs 0005, 0032, and 0052. In the Response to Arguments section of the Office Action mailed August 23, 2007, the Examiner states, "As suggested by Shafron, one of ordinary skill in the art would employ a DOM as a matter of design choice [0032]." These paragraphs describe scripts that may be used to manipulate a Document Object Model (DOM), such as for adding functionality to a web page. This has nothing to do with **generating a Document Object Model (DOM) tree** from each of a plurality of accessed configuration files.

Similar remarks apply also to claims 21, 50, and 67 which recite limitations involving generating one or more DOM trees comprising information from one or more configuration files; to claims 19, 57, 58, and 71, which recite limitations involving generating a DOM tree comprising information from a batch configuration document; and to claims 20, 22, 57, 59, and 71, which recite limitations involving applying information from a DOM tree to a configuration file associated with a component, all of which the Examiner submits are taught by Shafron in paragraph [0005], [0032], and

[0051]. Appellant asserts that these paragraphs do not teach all the limitations of these claims, as the Examiner's citations have nothing to do with generating or accessing DOM trees containing information from or for configuration files.

2. The Examiner has not established a proper motivation to combine the cited references to teach the limitations of these claims in his remarks regarding these claims.

For at least the reasons above, the rejection of claims 6, 19-22, 50, 57-59, 67, and 71 is unsupported by the cited art and removal thereof is respectfully requested.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-72 was erroneous, and reversal of his decision is respectfully requested.

Since this Appeal Brief is submitted within one month from the Notice of Panel Decision, no extension of time fee should be due. The Commissioner is authorized to charge the appeal brief fee and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-78701/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Appellant

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: December 14, 2007

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method for generating a batch configuration document for an intelligent device, the method comprising:

accessing a plurality of configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of a plurality of software components of the intelligent device; and

generating the batch configuration document from the plurality of configuration files, wherein the batch configuration document includes the configuration information for the plurality of software components of the intelligent device;

wherein, after said generating, the batch configuration document is accessible for use in configuring the plurality of software components of the intelligent device whose configuration files were used in said generating the batch configuration document.

2. The method as recited in claim 1, wherein said accessing the plurality of configuration files and said generating the batch configuration document are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for selecting the plurality of configuration files to be accessed and one or more executable instructions for performing said generating the batch configuration document.

3. The method as recited in claim 1, further comprising, prior to said accessing the plurality of configuration files, configuring the plurality of software

components of the intelligent device, wherein said configuring the plurality of software components sets the configuration information in the plurality of configuration files.

4. The method as recited in claim 1, further comprising transferring the batch configuration document to another intelligent device for use in configuring one or more software components of the other intelligent device.

5. The method as recited in claim 1, wherein the batch configuration document further includes configuration information for one or more software components of one or more other intelligent devices.

6. The method as recited in claim 1, wherein said generating the batch configuration document comprises generating a Document Object Model (DOM) tree from each of the accessed configuration files, wherein the configuration information incorporated in the configuration document is accessed from the DOM trees generated from the plurality of configuration files.

7. The method as recited in claim 1, further comprising configuring one or more of the plurality of software components of the intelligent device using the batch configuration document, wherein said configuring comprises applying the configuration information from the batch configuration document to one or more of the plurality of configuration files, wherein each of the one or more of the plurality of configuration files is associated with one of the one or more of the plurality of software components of the intelligent device.

8. The method as recited in claim 7, wherein said configuring the one or more of the plurality of software components of the intelligent device further comprises initializing each of the one or more of the plurality of software components, wherein said initializing uses the configuration information from the one or more configuration files associated with the particular component.

9. The method as recited in claim 1, wherein the plurality of software components includes software application programs.

10. The method as recited in claim 1, wherein the plurality of software components includes system software components.

11. The method as recited in claim 1, wherein the plurality of software components includes software drivers for hardware components.

12. The method as recited in claim 1, wherein at least one of the plurality of configuration files includes operating system configuration information for the intelligent device.

13. The method as recited in claim 1, wherein the batch configuration document is a markup language document.

14. The method as recited in claim 13, wherein the markup language is eXtensible Markup Language (XML).

15. The method as recited in claim 1, wherein the batch configuration document and the plurality of configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD).

16. A method for configuring a plurality of software components of an intelligent device, the method comprising:

accessing a batch configuration document, wherein the batch configuration document comprises configuration information for the plurality of software components of the intelligent device; and applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of the plurality of software components of the intelligent device.

17. The method as recited in claim 16, wherein said applying the configuration information from the batch configuration document to each of the one or more configuration files comprises replacing one or more current parameter values in the particular configuration file with new parameter values from the batch configuration document.

18. The method as recited in claim 16, wherein said accessing and said applying are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for accessing the batch configuration document and one or more executable instructions for selecting the one or more configuration files to be configured.

19. The method as recited in claim 16, wherein said accessing the batch configuration document comprises generating a Document Object Model (DOM) tree from the batch configuration document, wherein the DOM tree includes the configuration information for the one or more configuration files.

20. The method as recited in claim 19, wherein said applying the configuration information to the one or more configuration files comprises accessing the configuration information from the DOM tree generated from the batch configuration document.

21. The method as recited in claim 16, wherein said accessing the batch configuration document comprises generating a Document Object Model (DOM) tree for each of the one or more configuration files from the configuration information in the

batch configuration document, wherein each of the generated DOM trees comprises the configuration information for its associated configuration file.

22. The method as recited in claim 21, wherein, said applying the configuration information comprises:

for each of one or more of the plurality of software components of the intelligent device:

calling a module associated with the component;

passing a DOM tree generated from one of the one or more configuration files to the called module, wherein the configuration file is associated with the component, and wherein the DOM tree includes configuration information for the component; and

the called module applying the configuration information from the DOM tree to the configuration file associated with the component.

23. The method as recited in claim 16, wherein the plurality of software components includes one or more of software application programs.

24. The method as recited in claim 16, wherein the plurality of software components includes system software components.

25. The method as recited in claim 16, wherein the plurality of software components includes software drivers for hardware components.

26. The method as recited in claim 16, wherein at least one of the one or more configuration files includes operating system configuration information for the intelligent device.

27. The method as recited in claim 16, wherein the batch configuration document is a markup language document.

28. The method as recited in claim 27, wherein the markup language is eXtensible Markup Language (XML).

29. The method as recited in claim 16, wherein the batch configuration document and the one or more configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD).

30. The method as recited in claim 16, further comprising rebooting the intelligent device after said applying the configuration information from the batch configuration document to the one or more configuration files, wherein said rebooting applies the configuration information from the one or more configuration files to one or more of the plurality of software components of the intelligent device.

31. The method as recited in claim 16, further comprising initializing one or more of the plurality of software components of the intelligent device after said applying the configuration information from the batch configuration document to the one or more configuration files, wherein, in said initializing, each of the one or more of the plurality of software components is initialized using the configuration information from each of the one or more configuration files associated with the particular component.

32. The method as recited in claim 16, further comprising generating the batch configuration document on a different intelligent device prior to said accessing.

33. A method for configuring intelligent devices, the method comprising:

generating a batch configuration document from a plurality of configuration files on a first intelligent device, wherein each of the plurality of configuration files includes configuration information for one of one or more software components of the intelligent device, and wherein the batch configuration document includes the configuration information from the plurality of configuration files; and

configuring one or more software components of a second intelligent device using the batch configuration document generated on the first intelligent device.

34. The method as recited in claim 33, wherein said configuring comprises applying configuration information from the batch configuration document generated on the first intelligent device to one or more configuration files on the second device, wherein each of the one or more configuration files on the second intelligent device is associated with one of the one or more software components of the second intelligent device.

35. The method as recited in claim 33, wherein the batch configuration document is a markup language document.

36. The method as recited in claim 35, wherein the markup language is eXtensible Markup Language (XML), and wherein the batch configuration document and the plurality of configuration files conform to an XML Document Type Definition (DTD).

37. The method as recited in claim 33, wherein the one or more software components include one or more of software application programs, system software components, and software drivers for hardware components.

38. The method as recited in claim 33, wherein at least one of the plurality of configuration files on the first intelligent device includes operating system configuration

information for the first intelligent device, wherein the batch configuration document includes the operating system configuration information, and wherein said configuring the one or more software components of the second intelligent device comprises configuring an operating system of the second intelligent device using the operating system configuration information of the first intelligent device from the batch configuration document.

39. The method as recited in claim 33, further comprising rebooting the second intelligent device after said configuring, wherein said rebooting applies the configuration information from the batch configuration document to the one or more software components of the second intelligent device.

40. The method as recited in claim 33, further comprising:

storing the generated batch configuration document on a server, wherein the server is coupled to the second intelligent device via a network; and

downloading the stored batch configuration document to the second intelligent device;

wherein said configuring the one or more software components of the second intelligent device uses the downloaded batch configuration document.

41. A method for generating a batch configuration document for a plurality of intelligent devices, the method comprising:

accessing one or more configuration files on each of the plurality of intelligent devices, wherein each of the one or more configuration files on each of the plurality of intelligent devices includes configuration information for one of one or more software components of the intelligent device; and generating the batch configuration document from the one or more configuration files on each of the plurality of intelligent devices, wherein the batch configuration document includes the configuration information for the one or more software components of each of the plurality of intelligent devices;

wherein, after said generating, the batch configuration document is accessible for use in configuring the plurality of intelligent devices whose configuration files were used in said generating the batch configuration document.

42. The method as recited in claim 41, wherein the batch configuration document is further accessible for use in configuring other pluralities of intelligent devices.

43. The method as recited in claim 41, wherein said configuring the plurality of intelligent devices comprises applying the configuration information from the batch configuration document to the one or more configuration files for each of the one or more software components of each of the plurality of intelligent devices.

44. The method as recited in claim 41, wherein the one or more software components of each of the plurality of intelligent devices includes at least one of software application programs, system software components, and software drivers for hardware components.

45. The method as recited in claim 41, wherein at least one of the configuration files of at least one of the plurality of intelligent devices includes operating system configuration information for the intelligent device.

46. The method as recited in claim 41, wherein the batch configuration document is a markup language document.

47. The method as recited in claim 46, wherein the markup language is eXtensible Markup Language (XML), and wherein the batch configuration document and each of the one or more configuration files on each of the plurality of intelligent devices conform to an eXtensible Markup Language (XML) Document Type Definition (DTD).

48. An intelligent device, comprising:

a processor;

a plurality of software components;

a plurality of configuration files, wherein each of the plurality of configuration files is associated with one of the plurality of software components, and wherein each of the plurality of configuration files includes configuration information for its associated component; and

a memory operable to store program instructions, wherein the program instructions are executable by the processor to:

open each of the plurality of configuration files to access the configuration information for the component associated with the configuration file; and

generate a batch configuration document from the configuration information accessed from each of the plurality of configuration files;

wherein the batch configuration document includes the configuration information from each of the plurality of configuration files; and wherein, after said generating, the batch configuration document is accessible for use in configuring the plurality of software components in the intelligent device.

49. The intelligent device as recited in claim 48, wherein the batch configuration document is further transferable to another intelligent device comprising one or more software components similar to software components comprised in the plurality of software components of the intelligent device for use in configuring the other intelligent device.

50. The intelligent device as recited in claim 48, wherein, in said generating the batch configuration document, the program instructions are further executable by the processor to generate a Document Object Model (DOM) tree from each of the plurality of configuration files, wherein the configuration information included in the configuration document is accessed from the DOM trees generated from the plurality of configuration files.

51. The intelligent device as recited in claim 48, wherein, in said configuring the plurality of software components of the intelligent device, the program instructions are further executable by the processor to apply the configuration information from the batch configuration document to the one or more configuration files for each of the plurality of software components of the intelligent device.

52. The intelligent device as recited in claim 48, wherein the plurality of software components includes one or more of software application programs, system software components, and software drivers for hardware components.

53. The intelligent device as recited in claim 48, wherein the intelligent device further comprises an operating system for the intelligent device, and wherein at least one of the plurality of configuration files includes operating system configuration information for the intelligent device.

54. The intelligent device as recited in claim 48, wherein the batch configuration document is a markup language document.

55. The intelligent device as recited in claim 54, wherein the markup language is eXtensible Markup Language (XML), and wherein the batch configuration document and the plurality of configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD).

56. An intelligent device, comprising:

a processor;

a plurality of software components;

a plurality of configuration files, wherein each of the plurality of configuration files is associated with one of the plurality of software components, and wherein each of the plurality of configuration files includes configuration information for its associated component; and

a memory operable to store program instructions, wherein the program instructions are executable by the processor to:

open a batch configuration document, wherein the batch configuration document comprises configuration information for the plurality of software components of the intelligent device; and

apply the configuration information from the batch configuration document to the plurality of configuration files on the intelligent device.

57. The intelligent device as recited in claim 56, wherein the program instructions are further executable by the processor to:

generate a Document Object Model (DOM) tree from the batch configuration document, wherein the DOM tree includes the configuration information for the plurality of configuration files; and

wherein, in said applying the configuration information to the plurality of configuration files, the program instructions are further executable by the processor to access the configuration information from the DOM tree generated from the batch configuration document.

58. The intelligent device as recited in claim 56, wherein, in said accessing the batch configuration document, the program instructions are further executable by the processor to:

generate a Document Object Model (DOM) tree for each of the plurality of configuration files from the configuration information in the batch configuration document, wherein each of the generated DOM trees comprises the configuration information for its associated configuration file;

wherein the intelligent device further comprises a plurality of executable modules each associated with one of the plurality of software components, wherein each of the plurality of executable modules is operable to apply configuration information to the particular one of the plurality of configuration files associated with the component associated with the executable module.

59. The intelligent device as recited in claim 56, wherein, in said applying the configuration information to the plurality of configuration files, the program instructions are further executable by the processor to:

for each of the plurality of software components of the intelligent device:

call one of the plurality of modules, wherein the called module is associated with the component; and

pass a DOM tree generated from one of the plurality of configuration files to the called module, wherein the configuration file is associated with the component, and wherein the DOM tree includes configuration information for the component; and

wherein the called module is operable to apply the configuration information from the DOM tree to the configuration file associated with the component.

60. The intelligent device as recited in claim 56, wherein the plurality of software components includes one or more of software application programs, system software components, and software drivers for hardware components.

61. The intelligent device as recited in claim 56, wherein the intelligent device further comprises an operating system for the intelligent device, and wherein at least one of the plurality of configuration files includes operating system configuration information for the intelligent device.

62. The intelligent device as recited in claim 56, wherein the batch configuration document is a markup language document.

63. The intelligent device as recited in claim 62, wherein the markup language is eXtensible Markup Language (XML), and wherein the batch configuration document

and the one or more configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD).

64. The intelligent device as recited in claim 56, wherein the intelligent devices is operable to reboot after said configuring, wherein said rebooting applies the configuration information to the plurality of software components of the intelligent device.

65. The intelligent device as recited in claim 56, wherein the program instructions are further executable by the processor to initialize each of the plurality of software components of the intelligent device, wherein said initializing uses the configuration information from the one or more configuration files associated with the particular component.

66. A tangible, computer-accessible storage medium comprising program instructions, wherein the program instructions are computer-executable to implement:

accessing a plurality of configuration files on an intelligent device, wherein each of the plurality of configuration files includes configuration information for one of a plurality of software components of the intelligent device; and

generating a batch configuration document from the plurality of configuration files, wherein the batch configuration document includes the configuration information for the plurality of software components of the intelligent device;

wherein, after said generating, the batch configuration document is accessible for use in configuring the plurality of software components of the intelligent device whose configuration files were used in said generating the batch configuration document.

67. The tangible, computer-accessible storage medium as recited in claim 66, wherein, in said generating the batch configuration document, the program instructions are further computer-executable to implement generating a Document Object Model (DOM) tree from each of the plurality of accessed configuration files, wherein the configuration information incorporated in the configuration document is accessed from the DOM trees generated from the plurality of configuration files.

68. The tangible, computer-accessible storage medium as recited in claim 66, wherein the program instructions are further computer-executable to implement:

configuring one or more of the plurality of software components of the intelligent device using the batch configuration document;

wherein, in said configuring, the program instructions are further computer-executable to implement applying the configuration information from the batch configuration document to one or more of the plurality of configuration files, wherein each of the one or more of the plurality of configuration files is associated with one of the one or more of the plurality of software components of the intelligent device.

69. The tangible, computer-accessible storage medium as recited in claim 66, wherein the batch configuration document is a markup language document, wherein the markup language is eXtensible Markup Language (XML), and wherein the batch configuration document and the one or more configuration files conform to an XML Document Type Definition (DTD).

70. A tangible, computer-accessible storage medium comprising program instructions, wherein the program instructions are computer-executable to implement:

accessing a batch configuration document, wherein the batch configuration document comprises configuration information for a plurality of software components of an intelligent device; and

applying the configuration information from the batch configuration document to one or more configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of the plurality of software components of the intelligent device.

71. The tangible, computer-accessible storage medium as recited in claim 70, wherein, in said accessing the batch configuration document, the program instructions are computer-executable to implement:

generating a Document Object Model (DOM) tree from the batch configuration document, wherein the DOM tree includes the configuration information for the plurality of configuration files;

wherein, in said applying the configuration information to the one or more configuration files, the program instructions are computer-executable to implement accessing the configuration information from the DOM tree generated from the batch configuration document.

72. The tangible, computer-accessible storage medium as recited in claim 70, wherein the batch configuration document is a markup language document, wherein the markup language is eXtensible Markup Language (XML), and wherein the batch configuration document and the one or more configuration files conform to an XML Document Type Definition (DTD).

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.